UNITED STATES PATENT AND TRADEMARK OFFICE

A

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/091,645	03/05/2002	Handong Wu	NETAP019	9146

28875 7590 11/30/2005

Zilka-Kotab, PC
P.O. BOX 721120
SAN JOSE, CA 95172-1120

EXAMINER	
HOMAYOUNMEHR, FARID	

ART UNIT	PAPER NUMBER
2132	

DATE MAILED: 11/30/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/091,645	WU ET AL.	
	Examiner	Art Unit	
	Farid Homayounmehr	2132	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *10/12/2005*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-22* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-22* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *05 March 2002* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. The text of those sections of Title 35,U.S.Code not included in this section can be found in the prior office action.
2. The prior office actions are incorporated herein by reference. In particular, the observations with respect to claim language, and response to previously presented arguments.
3. Claims 1, 11, 19 and 20 were amended by the applicant.
4. Claims 21 and 22 are newly added by the applicant.
5. Claims 1-22 have been considered.
6. Claims 9, 12, 13 and 17 have been cancelled by the applicant.
7. Examiner withdraws the rejection under 35 U.S.C. 101 to claim 20 due to correction by the applicant. However, claim 20 remains rejected, as it is dependent on claim 19, which is rejected.

Response to Arguments

8. Applicant's arguments filed 10/12/2005 have been fully considered but they are not persuasive (see the following).

The applicant suggests that Vaidya (examiner's cited prior art) fails to disclose all elements of the claimed invention. This is because applicant believes that Vaidya's invention includes only one processor, an intrusion detection module (applicant's

item 14 in Fig. 1) only, without a separate data monitoring device (applicant's item 16), and therefore does not require the application programming interfaces claimed by the applicant.

One of the two basis of applicant's argument to distinguish applicant's claim invention from the cited prior art is on the separation between the data monitoring device and intrusion detection device.

However, the claim language, even after the latest amendments, does not make any distinct separation between the intrusion detection device and the data monitoring device, except for a separation on the functionalities of those two devices. Referring to figure 2 and applicant's description on page 8 line 11 to page 9 line 12, both the intrusion detection device and the data monitoring device are parts of element 18, which is clearly shown as one network intrusion detection and analysis system (NIDAS). Therefore, from the network point of view, separation of the intrusion detection device and the data monitoring device (within the NIDAS) makes no difference in the functionality of the claimed intrusion detection and analysis system, method or apparatus, and Vaidya's data collectors (item 10 in Fig. 1) completely disclose applicant's NIDAS.

Separation of functionalities between an intrusion detection element and a data monitor element is implied in several areas of Vaidya's invention disclosure:

8.1. Contrary to applicant's suggestion, Vaidya does not limit his invention to one processor only. Referring to Vaidya's figure 4, items 36, 34 and 38 are clearly separate modules responsible for performing separate functionalities. Therefore, the notion of using separate modules to perform separate functionalities is clearly obvious and well known to Vaidya and disclosed in his invention.

8.2. Referring to figure 4 again, Vaidya performs the functionality of applicant's data monitoring device and intrusion detection device in item 36, however, within item 36 it discloses item 40 as a separate device. Based on Vaidya column 8 line 40 to 55, packet information is extracted and stored in cache register 40. All data packet information, including MAC, IP, Transport and Application layer data is extracted and entered in cache register 40. As this action is explained in Fig. 5, it clearly shows a separate function of packet data extraction and collection and storage. In the following steps, depicted in separate procedures and figures 6 to 10 by Vaidya, the data collected from packets (stored in item 40) is used to perform intrusion detection. Vaidya clearly separates the functionality of data collection, as described in Fig 5, from intrusion detection, as described in Figures 6 to 10.

8.3. In addition, in Vaidya's claim 1, the method of detecting intrusion attempts is broken down to several steps, among which are the following separate steps:

- monitoring network traffic transmitted over said communications network for data addressed to one of the said network objects.
- executing at least one attack signature profile included in said subnet corresponding to said network object to determine if said data addressed to said network object is associated with a network intrusion attempt.

Therefore, applicant's 'separate data monitoring device and intrusion detection device' is disclosed by Vaidya.

The second basis of applicant's argument is that the use of API's to call different applications of data monitoring device from the intrusion detection device is not disclosed by Vaidya. In fact, applicant's specifications or claims (original or amended) do not point out any advantage in separation of the intrusion detection device and data monitoring device that is distinct from teaching of Vaidya, except for the use APIs to invoke certain functionalities.

However, using APIs as a means to transfer information between objects or elements of a distributed system is obvious and well known to a person skilled in the art and it would have been obvious to a person skilled in the art to use the APIs in order to build the intrusion detection system invented by Viadya. The motivation to do so would have been to take advantage of an already prepared

and well tested element to perform part of the required functionality (transfer of extracted packet information from Vaidya's item 40 to perform intrusion detection, as described in column 7 line 25 to 30).

Even though APIs have been widely know application development tools since before the time of claimed invention, the examiner has cited a reference to reaffirm the general use of APIs in development of Intrusion Detection systems (see Claim Rejections – 35 USC 103).

Claim Rejections - 35 USC § 112

9. Claims 21 and 22 are rejected under 35 U.S.C. 112, first paragraph, because the specification, while being enabling for general functionality of the APIs, does not reasonably provide enablement for the particular APIs cited in the claims. The specification does not enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make the invention commensurate in scope with these claims.

9.1. In claim 21 the API is claimed to have the form of frame_context_pointer_position. What is the exact implication of this form? Does it identify the parameters exchanged between objects? How is an API that is compliant with this form distinguished from the one that is not? What is the specific advantage of this form?

9.2. In claim 22, the API is requested to include at least one of frame_tcp_bridge, frame_udp_bridge, frame_ip_bridge, frame_http_bridge. Page 12 of the specification merely mentions the incision of these items as an example. No description on the specific functionality or how to use these elements is provided.

10. Claims 21 and 22 are rejected as failing to define the invention in the manner required by 35 U.S.C. 112, second paragraph.

The claim(s) are narrative in form and replete with indefinite and functional or operational language. The structure which goes to make up the device must be clearly and positively specified. The structure must be organized and correlated in such a manner as to present a complete operative device. The claim(s) must be in one sentence form only. Note the format of the claims in the patent(s) cited.

Claim Rejections - 35 USC § 103

11. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the

subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

12. Claims 1 to 19 are rejected under 35 U.S.C. 102(e) as anticipated by or, in the alternative, under 35 U.S.C. 103(a) as obvious over Vaidya (US Patent No. 6,279.113) and further in view of Porras (US Patent Application Publication number 2003/0101358 A1, filed 11/28/2001).

12.1. As per claims 1, Vaidya is directed to an intrusion detection and analysis system (Fig.1 column 5 lines 5 to 9) comprising:
a data monitoring device comprising a capture engine operable to capture data passing through the network (Fig. 1, item 10 column 5, lines 9 to 25) in response to a trigger (as per Fig. 3, the data monitor 58 continues to monitor data when triggered by item 64. see also column 7 line 4 to 6) and configured to monitor network traffic (Fig. 1, item 10 column 5, lines 13 to 15, see also Fig. 4 item 36 column 7 lines 11 to 15), decode protocols (Fig. 2, item 36 within item 10, column 7 lines 17 to 24), for grouping packets into different protocol presentations and assembling the packets into high level protocol groups (column 7 line 16 to 24 indicates that Vaidya determines the layer of OSI stack to which the monitored

packet belongs and performs intrusion detection based on profiles associated with OSI layers), and analyze received data (column 6 line 57 to column 7 line10) for managing the network by collecting statistics (column 8 lines 15 to 40 and column 9 lines 4 to 21 in conjunction with the response to claim 9 in the first office action show collection of statistics by Vaidya), and detecting broken lines, traffic loads, and network errors (as described in column 3 line 11 to 26, the attack signature profiles used by Vaidya's system to perform intrusion detection include profiles to detect data transport alteration or deletion, and malfunction of network objects. Broken lines cause deletion of transport data and malfunction of network objects causes network errors. Traffic load is measured by obtaining statistics on data packets traveling in the network. Hence Vaidya's system is capable of collecting statistics and detecting broken lines, traffic loads and network errors), and;

an intrusion detection device (Fig 2, item 36) separate from the data monitoring device (Vaidya's system obviously discloses the functionally separate data monitoring device and intrusion detection device. See section 8 above), the intrusion detection device comprising a detection engine operable to perform intrusion detection on data provided by the data monitoring device (item 36 column 6 lines 7 to 13);

application program interfaces configured to allow the intrusion detection device

access to applications of the data monitoring device to perform intrusion detection. Application program interface is defined as 'a set of functions or methods used to access some functionality'. Vaidya does not explicitly mention the use of APIs, however, referring to Fig. 2, the configuration builder module (item 32) allows the intrusion detection device (item 36) access attack signature profiles stored in signature profile memory (item 39), as described in column 6 lines 1 to 11. Also, the communication module (item 34) allows intrusion detection device access the data in database handler (item 26). This clearly allows the intrusion detection device access the functionality of the data-monitoring device to perform intrusion detection, and hence discloses the feature. Therefore, the Examiner asserts that Vaidya discloses the feature by inheritance.

Vaidya does not specifically refer to an application program interface to access the data and functionalities of the data-monitoring device, but Porras clearly suggests the use of APIs (paragraphs 40 to 42 and claim 2) in development of Intrusion Detection Systems. It would have been obvious to a person skilled in art to use APIs, as clearly disclosed by Porrass, as a means to transfer information between objects or elements of a distributed system in order to build the intrusion detection system invented by Viadya. The motivation to do so would have been to take advantage of an already prepared and well tested element to perform part of the required functionality (transfer of extracted packet information

from Vaidya's item 40 to perform intrusion detection, as described in column 7 line 25 to 30).

memory for storing reference network information used by the intrusion detection device to determine if an intrusion has occurred (Fig 2, item 26, column 5 lines 46 to 50);

wherein the application program interfaces allow the intrusion detection device to leverage the separate data monitoring device, by allowing the intrusion detection device to call an application programming interface configured to open protocol decoding application associated with the separate data monitoring device and by allowing the intrusion detection device to call and API configured to open an alarm generation application associated with the separate data monitoring device (as mentioned above Vaidya discloses APIs, by the way of a set of functions or methods to access some functionality. Vaidya discloses the functionality of protocol decoding as described above and in column 7 line 16 to 24. Vaidya also discloses the functionality of alarm generation, as described in rejection of claim 13 in the first office action. Therefore, Vaidya discloses the entire feature).

12.2. As per claim 2, Vaidya continues to teach a reference network information comprises a signature database including signature profiles associated with a known network security violation (Fig 2, item 39 column 6 lines 1 to 7) and wherein the detection engine (item 32) is operable to compare the data provided

by the data monitoring device (item 39) with the signature profiles to detect network intrusions (column 6 lines 7 to 18).

12.3. As per claim 3, Vaidya is directed to a parser (Fig. 4 item 36) operable to parse (referring to Fig. 8 and column 9 lines 45 to 55, and Fig. 9 and column 10 line 17 to column 11 line 15, the data collector, as part of processing the attack signature profile, separates the fields in the signature profile, which is equivalent to parsing), generate (Fig. 2 and 3 and column 6 lines 27 to 29), and load (Fig. 2 column 6 lines 1 to 11 and Fig. 4 column 7 lines 18 to 23) signatures at the detection engine.

12.4. As per claim 4, Vaidya is directed to the reference network information comprises a baseline state of network traffic (state cache as shown in Fig. 4 item 44 and described in column 9 lines 3 to 20 stores the network traffic state data) and wherein the detect engine is operable to compare the data received by the capture engine to the baseline network state and look for anomalies (claims 11 and 12).

12.5. As per claim 5, Vaidya is directed to a data-monitoring device in accordance with claim 4 that provides the baseline state of network traffic. Vaidya provides the baseline state of network traffic in the virtual processor (Fig. 4 item 36), wherein the Register Cache stores the information extracted from a

packet (as described in column 7 lines 11 to 17), and in the case where a sequential or timer/counter based signature profiles are invoked (column 7 lines 48 to 51), the state of network traffic (data extracted from the packet) is saved in the state cache (Fig 4. item 44) and used in conjunction with the data from the subsequent packet (column 7 lines 59 to 65) to determine network anomalies. The use of traffic state data to detect network anomalies is further described in column 8 lines 16 to 40.

12.6. As per claim 6, Vaidya is directed to a log file configured to at least temporarily store reports generated by the detect engine, on account of the Reaction Module (Fig. 2 item 38), which receives alerts from the detection device whenever a network intrusions is detected, and initiates reactions depending on nature of the attack (column 6 lines 18 to 26).

12.7. As per claim 7, Vaidya is directed to a system according to claim 6, further comprising an alarm manager operable to generate alarms based on the information in the log file (column 6 lines 21 to 26).

12.8. As per claim 8, Vaidya discloses a system in accordance with claim 1 further comprising a filter configured to filter out packets received at the data-monitoring device. Filter is defined as 'a device that separates data in accordance with specific criteria'. Item 96 in Fig. 6 (which describes the

operation of the data collector) returns 'No Entry Found' when the data collected from the packet indicates that the packet's destination server is not being monitored, and no further action will take place on that packet (column 8 lines 58 to 65). This clearly discloses 'a separation of packets based on a criteria' and hence discloses a 'filter'. Vaidya does not specifically refer to a filter. Therefore, the Examiner asserts that it Vaidya discloses the feature.

12.9. Claim 9 has been cancelled by the applicant.

12.10. As per claim 10, Vaidya is directed to a system in accordance with claim 1 wherein the capture engine is configured to forward packets and temporarily store packets for later analysis by data monitoring device (Fig. 4 item 40 column 7 lines 15 to 24).

12.11. As per claim 11, Vaidya is directed to a method for performing intrusion detection with an intrusion detection and analysis system (Fig.1 and column 5, as described in lines 5 to 9) comprising a data monitoring device including a capture engine operable to capture data passing through the network in response to a trigger (as per Fig. 3, the data monitor 58 continues to monitor data when triggered by item 64. see also column 7 line 4 to 6) and configured to monitor network traffic (Fig. 1, item 10 column 5, lines 13 to 15, see also Fig. 4 item 36 column 7 lines 11 to 15), decode protocols (Fig. 2, item 36 within item 10, column

7 lines 17 to 24), for grouping packets into different protocol presentations and assembling the packets into different protocol presentations and assembling the packets into high level protocol groups (column 7 line 16 to 24 indicates that Vaidya determines the layer of OSI stack to which the monitored packet belongs and performs intrusion detection based on profiles associated with OSI layers), and analyze received data (column 6 line 57 to column 7 line10) for managing the network by collecting statistics (column 8 lines 15 to 40 and column 9 lines 4 to 21 in conjunction with the response to claim 9 in the first office action show collection of statistics by Vaidya), and detecting broken lines, traffic loads, and network errors (as described in column 3 line 11 to 26, the attack signature profiles used by Vaidya's system to perform intrusion detection include profiles to detect data transport alteration or deletion, and malfunction of network objects. Broken lines cause deletion of transport data and malfunction of network objects causes network errors. Traffic load is measured by obtaining statistics on data packets traveling in the network. Hence Vaidya's system is capable of collecting statistics and detecting broken lines, traffic loads and network errors), and an intrusion detection device (Fig 2, item 36) separate from the data monitoring device (Vaidya's system obviously discloses the functionally separate data monitoring device and intrusion detection device. See section 8 above), coupled to data monitoring device (Fig. 2 items 34 and 30 couple the intrusion detection device to the data collected by the data monitoring device) and configured to perform intrusion detection on data provided by the data monitoring device

(column 6 lines 1 to 21); the method comprising:

receiving data at data monitoring device (Fig 4. item 36 column 7 lines 18 to 24);

capturing at least a portion of the packets contained within the data (column 7 lines 18 to 24);

by allowing the intrusion detection device to call at least one application program interface configured to open application of the data monitoring device; and performing intrusion detection at the intrusion detection device utilizing at least one of the applications of the data monitoring device (Application program interface is defined as 'a set of functions or methods used to access some functionality'. Referring to Fig. 2, the configuration builder module (item 32) allows the intrusion detection device (item 36) access attack signature profiles stored in signature profile memory (item 39), as described in column 6 lines 1 to 11. Also, the communication module (item 34) allows intrusion detection device access the data in database handler (item 26). This clearly allows the intrusion detection device access the functionality of the data-monitoring device to perform intrusion detection, and hence discloses the feature.

Vaidya does not specifically refer to an application program interface to access the data and functionalities of the data-monitoring device, but Porras clearly

suggests the use of APIs (paragraphs 40 to 42 and claim 2) in development of Intrusion Detection Systems. It would have been obvious to a person skilled in art to use APIs, as clearly disclosed by Porrass, as a means to transfer information between objects or elements of a distributed system in order to build the intrusion detection system invented by Viadya. The motivation to do so would have been to take advantage of an already prepared and well tested element to perform part of the required functionality (transfer of extracted packet information from Vaidya's item 40 to perform intrusion detection, as described in column 7 line 25 to 30).

wherein the application program interfaces allow the intrusion detection device to leverage the separate data monitoring device, by allowing the intrusion detection device to call an application programming interface configured to open protocol decoding application associated with the separate data monitoring device and by allowing the intrusion detection device to call and API configured to open an alarm generation application associated with the separate data monitoring device (as mentioned above Vaidya discloses APIs, by the way of a set of functions or methods to access some functionality. Vaidya discloses the functionality of protocol decoding as described above and in column 7 line 16 to 24. Vaidya also discloses the functionality of alarm generation, as described in rejection of claim 13 in the first office action. Therefore, Vaidya discloses the entire feature).

12.12. Claim 12 is cancelled by the applicant.

12.13. Claim 13 is cancelled by the applicant.

12.14. As per claim 14, Vaidya discloses a method of claim 11 further comprising filtering prior to capturing packets. Filtering is defined as 'a method to separate data in accordance with specific criteria'. Item 96 in Fig. 6 (which describes the operation of the data collector) returns 'No Entry Found' when the data collected from the packet indicates that the packet's destination server is not being monitored, and no further action will take place on that packet (column 8 lines 58 to 65). This clearly discloses 'a separation of packets based on a criteria' and hence discloses 'filtering'. Vaidya does not specifically refer to filtering. Therefore, the Examiner asserts that Vaidya discloses the feature.

12.15. As per claim 15, Vaidya discloses a method of claim 11 wherein performing intrusion detection comprises performing signature matching (Fig. 4 column 7 line 31 to column 8 line 40 and claim 1).

12.16. As per claim 16, Vaidya discloses a method of claim 15 wherein the application program interfaces provide parsing of signatures used in signature matching (column 10 lines 17 to 45).

12.17. Claim 17 is cancelled by the applicant.

12.18. As per claim 18, Vaidya discloses a method of claim 11 wherein performing intrusion detection comprises detecting anomalies in the received data (column 6 line 57 to column 7 line10).

12.19. As per claim 19, Vaidya discloses a computer program product (in form of a set of instructions to be sequentially executed as described in column 16 claim 18) for performing intrusion detection with an intrusion detection and analysis system comprising a data monitoring device including a capture engine operable to capture data passing through the network in response to a trigger (as per Fig. 3, the data monitor 58 continues to monitor data when triggered by item 64. see also column 7 line 4 to 6) and configured to monitor network traffic (Fig. 1 column 5 lines 5 to 9) comprising a data monitoring device configured to monitor network traffic (Fig. 1, item 10 column 5, lines 13 to 15, see also Fig. 4 item 36 column 7 lines 11 to 15), decode protocols (Fig. 2, item 36 within item 10, column 7 lines 17 to 24) for grouping packets into different protocol presentations and assembling the packets into different protocol presentations and assembling the packets into high level protocol groups (column 7 line 16 to 24 indicates that Vaidya determines the layer of OSI stack to which the monitored packet belongs and performs intrusion detection based on profiles associated with OSI layers),

and analyze received data (column 6 line 57 to column 7 line10) for managing the network by collecting statistics (column 8 lines 15 to 40 and column 9 lines 4 to 21 in conjunction with the response to claim 9 in the first office action show collection of statistics by Vaidya), and detecting broken lines, traffic loads, and network errors (as described in column 3 line 11 to 26, the attack signature profiles used by Vaidya's system to perform intrusion detection include profiles to detect data transport alteration or deletion, and malfunction of network objects. Broken lines cause deletion of transport data and malfunction of network objects causes network errors. Traffic load is measured by obtaining statistics on data packets traveling in the network. Hence Vaidya's system is capable of collecting statistics and detecting broken lines, traffic loads and network errors), and an intrusion detection device (Fig 2, item 36) separate from the data monitoring device (Vaidya's system obviously discloses the functionally separate data monitoring device and intrusion detection device. See section 8 above), coupled to data monitoring device (Fig. 2 items 34 and 30 couple the intrusion detection device to the data collected by the data monitoring device) and configured to perform intrusion detection on data provided by the data monitoring device (column 6 lines 1 to 21); the product comprising:

code that receives data at data monitoring device (Fig 4. item 36 column 7 lines 18 to 24);

code that captures at least a portion of the packets contained within the data (column 7 lines 18 to 24);

code that calls at least one application program interface configured to open application of the data monitoring device; and performs intrusion detection at the intrusion detection device utilizing at least one of the applications of the data monitoring device. (Application program interface is defined as 'a set of functions or methods used to access some functionality'. Referring to Fig. 2, the configuration builder module (item 32) allows the intrusion detection device (item 36) access attack signature profiles stored in signature profile memory (item 39), as described in column 6 lines 1 to 11. Also, the communication module (item 34) allows intrusion detection device access the data in database handler (item 26). This clearly allows the intrusion detection device access the functionality of the data monitoring device to perform intrusion detection, and hence discloses the feature. (Vaidya does not specifically refer to an application program interface to access the data and functionalities of the data-monitoring device, but Porras clearly suggests the use of APIs (paragraphs 40 to 42 and claim 2) in development of Intrusion Detection Systems. It would have been obvious to a person skilled in art to use APIs, as clearly disclosed by Porrass, as a means to transfer information between objects or elements of a distributed system in order to build the intrusion detection system invented by Viadya. The motivation to do so would have been to take advantage of an already prepared and well tested

element to perform part of the required functionality (transfer of extracted packet information from Vaidya's item 40 to perform intrusion detection, as described in column 7 line 25 to 30)

; and

a computer-readable storage medium for storing codes (Fig. 2 item 39 and associated description).

wherein the application program interfaces allow the intrusion detection device to leverage the separate data monitoring device, by allowing the intrusion detection device to call an application programming interface configured to open protocol decoding application associated with the separate data monitoring device and by allowing the intrusion detection device to call and API configured to open an alarm generation application associated with the separate data monitoring device (as mentioned above Vaidya discloses APIs, by the way of a set of functions or methods to access some functionality. Vaidya discloses the functionality of protocol decoding as described above and in column 7 line 16 to 24. Vaidya also discloses the functionality of alarm generation, as described in rejection of claim 13 in the first office action. Therefore, Vaidya discloses the entire feature).

12.20. Claim 20 is rejected because it depends on claim 19 which is rejected.

Conclusion

13. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

14. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Farid Homayounmehr whose telephone number is (571) 272-3937. The examiner can be normally reached on 9 hrs Mon-Fri, off Monday biweekly.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Gilberto Barron can be reached on (571) 272-3799. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Farid Homayounmehr

11/1/2005

GILBERTO BARRON JR.
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100